アセンブリ言語(超簡単命令セット版)

取扱説明書

－平成26年2月改訂版－

目次

1 はじめに

このアプリケーションは「計算機ハードウェア論」のアセンブリ言語(超簡単命令セット)の理解を助けるために製作されました。便宜的に機能を追加・削除した箇所があるため、このアプリケーション上での動き方が実際のCPUでの動き方と異なる場合があることに留意してください。このアプリケーションでは以下の事柄をシミュレーションできます。

- プログラムの実行
- アセンブリ言語の機械語への変換
- メモリアクセスの速度の体感

ユーザーはアセンブリ言語(超簡単命令セット版)に関する基本的な知識があることを前提としています。このマニュアルで分からない単語が出てきた場合は他の資料を参照してください。

このアプリケーションはJavaの環境が必須ですので、プラグインを有効にしてください。さらに、Tabキーを使用する場合は「Java7Update51バージョン」にして操作してください。

1.1 利用できるデータの範囲やアドレス空間、命令長についての注意

利用できるデータの範囲やアドレス空間、命令長に関して以下のように定めています。

- 利用可能なデータ型は8ビット符号つき整数型(-128 ~ 127)
- アドレス空間は8ビット
- 命令長は2バイト
- 余ったビットは0詰め

2 プログラムの書き方と実行の方法

2.1 このマニュアルでの表記について

ニーモニック中の記号は次のような意味です。

#n 整数値を表します。これが書かれているオペランドでは次のような表記ができます。

・ 直接数字を書く方法

#のすぐ後ろに10進数で数字を書いてください。例）#0,#-100

この場合、#nは単なる数字を表しています。

・ C言語のchar型を利用する方法

#のすぐ後ろに'（シングルクオーテーション） で囲んで文字を1字書いてください。

エスケープシーケンスは進数変換を表すもの以外は使用できます。例）#'a',#'\n' この場合、#nはその文字のASCIIコードを表しています。

・ ラベルを利用する方法

#を書かずにラベル名をそのまま書いてください。例）NEXT,i

この場合、#nはラベルの書かれているアドレスを表しています。

address そのラベル名が書かれているアドレス値を表しています。これが書かれているオペランドではラベルを指定してください。例） i,RET

Rd/Rs/Ra/Rb レジスタを表しています。レジスタは4 個使用でき、その名前はR0/R1/R2/R3です。

M() メモリを表しています。()内にレジスタまたは即値やラベルを指定することで、そのアドレスの値を呼び出すことができます。例） M(R0),M(R1+i),M(R2+#2)

2.2 入力上の注意

2.2.1 使用できる文字について

プログラムに使用できる文字は、半角英数記号です。全角文字は使用できません。

2.2.2 ラベルの付け方

各行の左端のラベル欄にラベルを指定することができます。ラベル名は100字以内としてください。また日本語のラベルを付けることも可能です。例） i,RET,終わり

2.3 命令長の書き方

命令は16通りあります。以下にそれぞれの命令長の書き方と説明を記載します。

LOAD 命令

レジスタに値を格納する命令です。

LOADI Rd #n	Rd に即値n の値を格納します。
LOADA Rd M(address)	Rd にaddress で示すメモリに入っている値を格納します。
LOADX Rd M(Rs+#n)	Rd にRs に入っている値とn の値を足した値をアドレスとしたメモリに入っている値を格納します。M(Rs+#0) を省略してM(Rs) と書くこともできます。配列や構造体を実装するときに使用します。 「コンピュータアーキテクチャ入門（城和貴著）」の表記に合わせるため、メモリ内の表記は(Rs+#n)の順番でのみ取り扱うものとします。

STORE 命令

メモリにレジスタの値を格納する命令です。

STOREA M(address) Rs	address で示すメモリにRs に入っている値を格納します。
STOREX M(Rd+#n) Rs	Rs に入っている値とn の値を足した値をアドレスとしたメモリにRs に入っている値を格納します。M(Rd+#0) を省略してM(Rd) と書くこともできます。 配列や構造体を実装するときに使用します。 「コンピュータアーキテクチャ入門（城和貴著）」の表記に合わせるため、メモリ内の表記は(Rs+#n)の順番でのみ取り扱うものとします。

算術演算命令

算術演算を行う命令です。

ADD Rd Rs	Rd にRd に入っている値とRsに入っている値を足した値を格納します。
SUB Rd Rs	Rd にRd に入っている値からRsに入っている値を引いた値を格納します。
INC Rd	Rd に入っている値を1 増やしてRd に格納します。
DEC Rd	Rd に入っている値を1 減らしてRd に格納します。

条件分岐命令

条件に合っていれば指定されたアドレスに飛ぶ命令です。

BRGT Ra Rb address	Ra がRb よりも大きければaddress で示すアドレスに飛びます。

BREQ Ra Rb address　　　Ra がRb と等しければaddress で示すアドレスに飛びます。
BRLT Ra Rb address　　　Ra がRb よりも小さければaddress で示すアドレスに飛びます。

無条件分岐命令

無条件に指定された行に飛ぶ命令です。

BRA address　　無条件にaddress で示すアドレスに飛びます。
BRR Rd　　Rd に入っている値をアドレスとし、無条件にそのアドレスに飛びます。

CALL 命令

決められた関数を呼び出す命令です。

CALL DISPLAY Rs　　Rs に入っているASCII コードの値を実行出力用フィールドに書き出します。ここで、Rs にはASCII コードの値が入っていることが前提のため、値を出力するためにはASCII コード化の処理をすることが求められます。
CALL INPUT Rd　　実行入力用フィールドから1 文字読み取り、それをASCII コードにしてRd に格納します。

その他

HLT　　この行でプログラムを終了します。終了したい行に必ず書いてください。

2.3.1 メモリの記入について

メモリはメモリ入力欄に記入してください。ラベルはオペランド１の欄に、初期値を入力したいときはオペランド２の位置に初期値を入力してください。初期値には数字または文字列を指定できます。数字を指定するときはそのまま数字を書き、文字列を指定するときは”（ダブルクオーテーション）で囲んだ中に文字列を書いてください。メモリの範囲では空行はラベル・初期値なしのメモリとして解釈されます。

2.3.2 空白・コメント文について

このプログラムでは空行、コメント文が使えます。ただし、空行が指定できるのはプログラム中のみです。メモリの中では空行も一行としてみなされます。コメントは入力画面1(p.7)に入力する場合はコメント欄に、入力画面2(p.9)に入力する場合は「//」の後に記入してください

2.3.3 アセンブリ言語と機械語の対応

それぞれのオペコード、レジスタは以下のように対応します。

・オペコード

アセンブリ	機械語	アセンブリ	機械語	アセンブリ	機械語	アセンブリ	機械語
LOADI	0000	LOADA	0001	LOADX	0010	STOREA	0011
STOREX	0100	ADD	0101	SUB	0110	INC	0111
DEC	1000	BRGT	1001	BREQ	1010	BRLT	1011
BRA	1100	BRR	1101	CALL	1110	HLT	1111

・CALL の関数

アセンブリ	機械語	アセンブリ	機械語
DISPLAY	00	INPUT	01

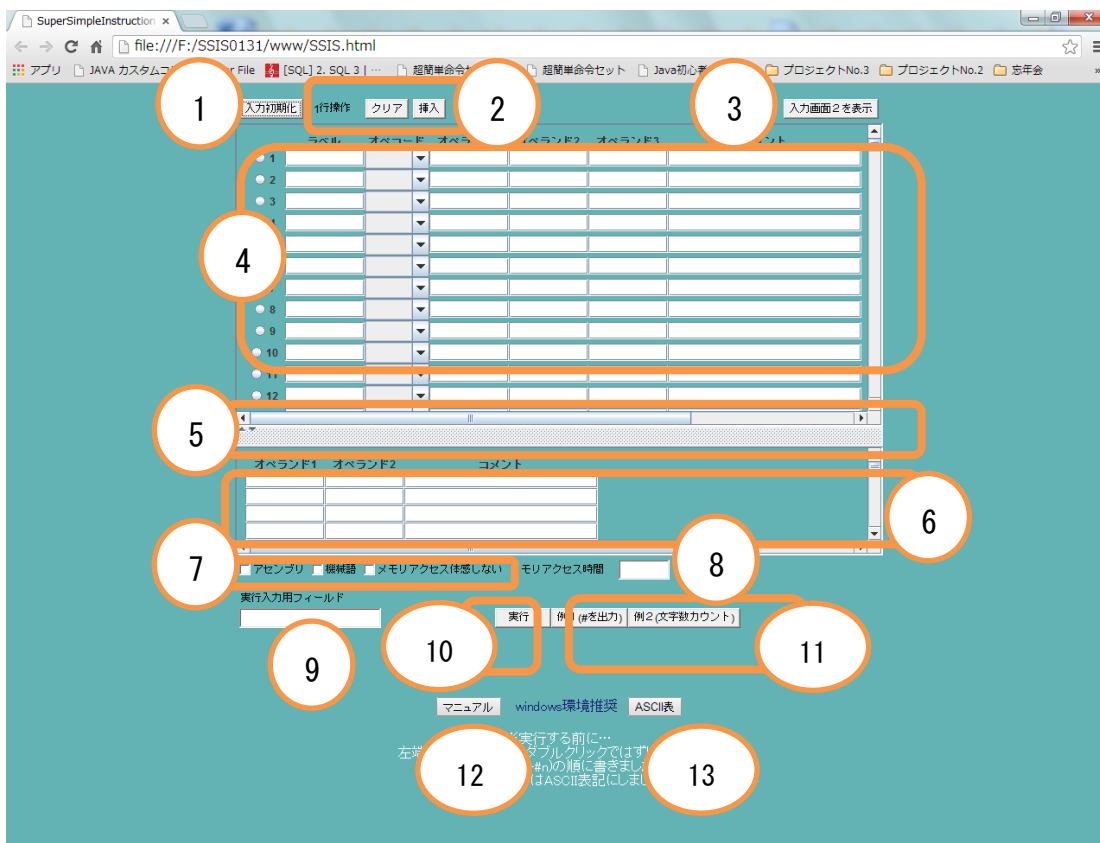
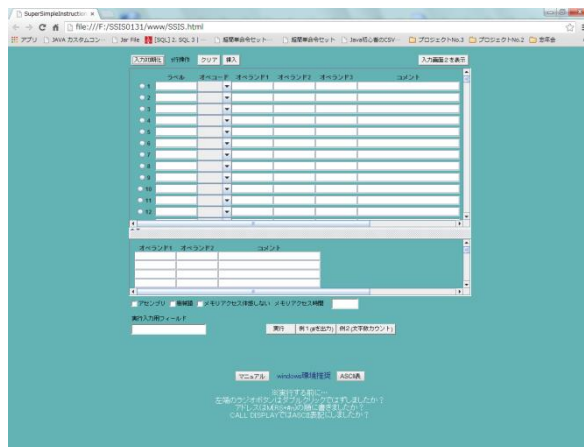
・レジスタ

アセンブリ	機械語	アセンブリ	機械語	アセンブリ	機械語	アセンブリ	機械語
R0	00	R1	01	R2	10	R3	11

2.4 実行方法

2.4.1 入力画面

2.4.1.1 入力画面1

命令調入力欄にプログラムを入力し、実行ボタンをクリックしてください。

1. **入力初期化ボタン**

命令入力欄を初期化します。

2. **一行操作ボタン**

- クリアボタン…ラジオボタンで選択された行を初期化します。
- 挿入ボタン…ラジオボタンで選択された行の上に空行を挿入します。

3. **入力画面２表示ボタン**

入力画面２（p.9）を表示します。

4. **命令入力欄**

アセンブリ言語のプログラムを入力してください。

- ラジオボタン…行を選択します。

5. **ディバイダ**

命令入力欄とメモリ入力欄の幅（命令修正数とメモリ修正数）を調節することができます。

6. **メモリ入力欄**

メモリを入力してください。

7. **チェックボックス**

- アセンブリ…チェックして実行すると、アセンブリ出力画面（p.11）が開き、命令入力欄とメモリ入力欄に入力されたアセンブリ言語が出力されます。
- 機械語…チェックして実行すると、機械語出力画面（p.11）が開き、命令入力欄とメモリ入力欄に入力されたアセンブリ言語を機械語に変換したものが出力されます。
- メモリアクセス体感しない…チェックして実行すると、メモリアクセス時間として停止している時間をなくします。チェックしない場合、メモリアクセス時間入力欄に入力された時間だけプログラムを停止します（→p.12）。

8. **メモリアクセス時間入力欄**

メモリアクセス時間を半角の整数で入力してください。入力しなかった場合は200に設定されます。単位はミリ秒です。

9. **実行入力用フィールド**

CALL INPUTで読み込むためのフィールドです。入力するときは１文字ごとに空白を入れてください。

10. **実行ボタン**

プログラムを実行します。実行結果は結果出力画面（p. 10）に表示します。

11. **例ボタン**

二重ループ（例１）と配列を使う（例２）プログラム例がそれぞれ記入されています。入力画面２が起動されるので、アセンブリ読み込みボタンを押すことで入力画面１にプログラムが反映されます。

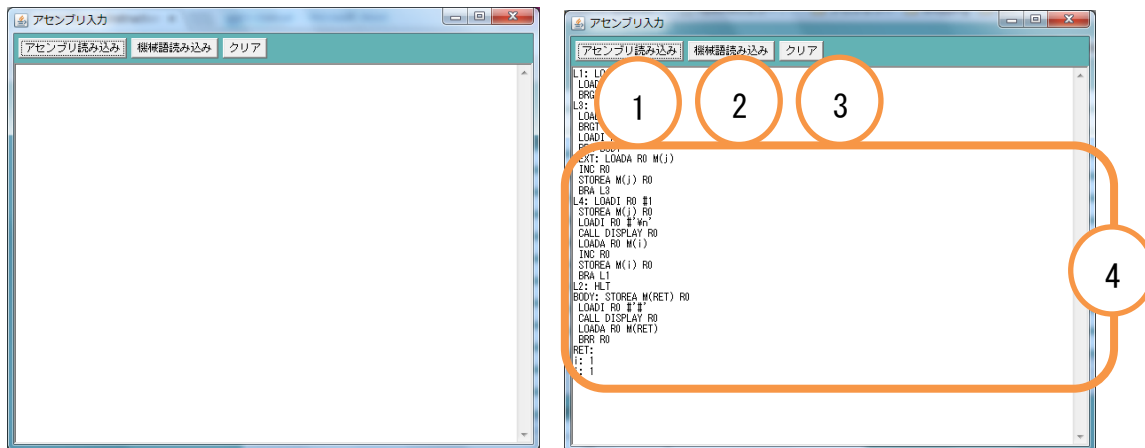
12. **マニュアルボタン**

マニュアルページに飛びます。

13. **ASCII表ボタン**

ASCII表が表示されます。

2.4.1.2 入力画面2

入力画面１で入力画面２表示ボタンを押すと表示されます。

1. **アセンブリ読み込みボタン**

プログラム入力欄に入力されたアセンブリ言語を入力画面１（p. 7）に反映させます。

2. **機械語読み込みボタン**

プログラム入力欄に入力された機械語をアセンブリ言語に変換後、入力画面１に反映させせます。

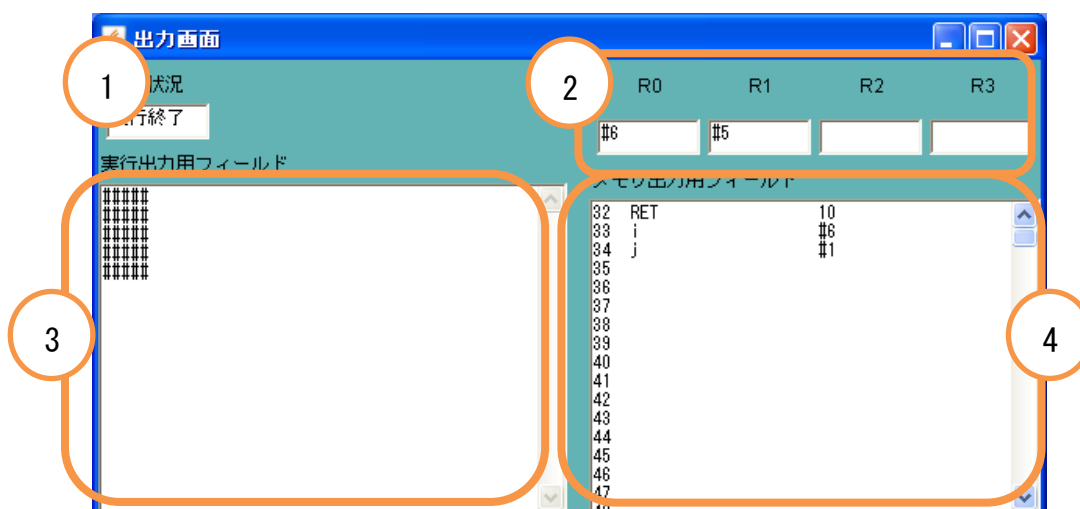
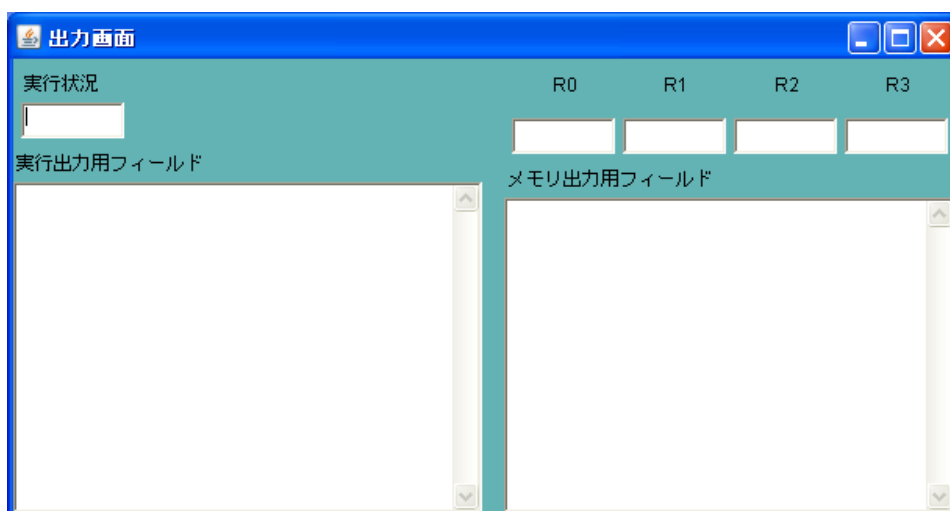
3. **クリアボタン**

プログラム入力欄を初期化します。

4. **プログラム入力欄**

アセンブリ言語、または機械語のプログラムを入力してください。

2.4.2. 結果出力画面

入力画面１で実行ボタンを押すと表示されます。

1. **実行状況フィールド**

   プログラムが実行中か、実行終了かを表示します。

2. **レジスタフィールド**

   現在の各レジスタの状態を表示します。

3. **実行出力フィールド**

   エラー文と、CALL DISPLAYによる出力を表示します。

4. **メモリ出力用フィールド**

   現在のメモリの状態を表示します。左からアドレス、ラベル、値の順で表示します。

2.4.3 アセンブリ出力画面

入力画面１のアセンブリにチェックして実行ボタンを押すと表示されます。機械語にもチェックした場合、アセンブリ言語は新たに表示された画面の左のフィールドに表示されます。

2.4.4 機械語出力画面

入力画面１の機械語にチェックして実行ボタンを押すと表示されます。アセンブリにもチェックした場合、機械語は新たに表示された画面の右のフィールドに表示されます。

2.5 機械語 の表記について

機械語は左からアドレス、オペコード、オペランドの順に出力されます。オペコードやオペランドの間は空白で区切って出力します。アセンブリ言語と機械語の対応は2.3.3.アセンブリ言語と機械語の対応（p.6）を参照してください。

3 メモリアクセス時間について

3.1 チェックボックス”メモリアクセス体感しない” に関する注意

このプログラムではメモリアクセスがレジスタに比べてかなりの時間を要することを確認できるよう、人間が体感できる程度の時間をとっています。実際のメモリアクセスはこのプログラムほど時間がかからないことに留意してください。

4 プログラム例

以下に二重ループと配列を使うプログラム例を載せますので参考にしてください。その他のプログラムに関しては他の資料を参照してください。

4.1 二重ループを用いたプログラム

基本的な二重ループを使用した5 × 5 個の’#’を出力するプログラムです。サブルーチンBODY の内容を変えることによって様々な二重ループのプログラムが実装できます。入力画面２に入力するときには、ラベルの最後に:を、コメントの前に//を付け加えてください。

ラベル	オペコード	オペランド1	オペランド2	オペランド3	コメント
L1	LOADA	R0	M(i)		外側ループ
	LOADI	R1	#5		
	BRGT	R0	R1	L2	
L3	LOADA	R0	M(j)		内側ループ
	LOADI	R1	#5		
	BRGT	R0	R1	L4	
	LOADI	R0	NEXT		
	BRA	BODY			二重ループの中のボディー部分
NEXT	LOADA	R0	M(j)		
	INC	R0			
	STOREA	M(j)	R0		
	BRA	L3			
L4	LOADI	R0	#1		
	STOREA	M(j)	R0		
	LOADI	R0	#'\n'		
	CALL	DISPLAY	R0		
	LOADA	R0	M(i)		
	INC	R0			
	STOREA	M(i)	R0		
	BRA	L1			
L2	HLT				
BODY	STOREA	M(RET)	R0		サブルーチン
	LOADI	R0	#'#'		
	CALL	DISPLAY	R0		
	LOADA	R0	M(RET)		
	BRR	R0			

オペランド 1	オペランド 2	コメント
RET		BODY の後に実行するアドレスを格納
i	1	
j	1	

4.2 文字列の長さを調べるプログラム

文字列の長さを数えて出力するプログラムです。MSG の先頭アドレスから空文字 のアドレスまで、アドレスを後ろにひとつずつずらしながら数えています。配列を使うプログラムにも応用できます。

ラベル	オペコード	オペランド1	オペランド2	オペランド3	コメント
	LOADI	R0	NEXT		
	LOADI	R1	MSG		MSG の先頭アドレスを代入
	BRA	Count			
NEXT	LOADI	R2	#'0'		
	ADD	R2	R0		
	CALL	DISPLAY	R2		文字列の長さを出力
	HLT				
Count	STOREA	M(RET)	R0		サブルーチン。R0に長さを格納
	LOADI	R3	#'\0'		
	LOADI	R0	#0		
L2	LOADX	R2	M(R1)		
	BREQ	R2	R3	L1	
	INC	R0			
	INC	R1			
	BRA	L2			
L1	LOADA	R1	M(RET)		
	BRR	R1			

オペランド１	オペランド２	コメント
RET		Countの後に実行するアドレスを格納
MSG	"Hello"	文字列

5 Q&A（エラーとその対処法）

以下にエラー表示とその対処法を示します。エラーの原因が見つからない場合は、同じ行の前後にエラーがないか確認してください。

5.1 レジスタに関するエラー

「レジスタが正しくありません」	レジスタはR0～R3を指定してください。
「レジスタの要素がありません」	レジスタに要素が入っているか確認してください。

5.2 アドレスに関するエラー

「アドレスが正しくありません」	アドレスが正しく指定されているか確認してください。
「使用できるアドレスは0～255です」	アドレスが正しく指定されているか確認してください。
「アドレスは降順でかぶらないように書いてください」	アドレスが正しく指定されているか確認してください。

5.3 入力に関係するエラー

「入力文字がありません」	実行入力用フィールドに文字を入力してください。
「文字の間にはスペースを入れてください」	実行入力用フィールドの文字列には１文字ごとに空白を入れてください。
「文字入力が正しくありません」	入力した文字の形式が正しいかを確認してください。文字は、ASCIIコードに変換できるものを入力してください。
「メモリに入力できるのは文字列か数字です」	文字列か数字を入力してください。
「負の数は出力できません」	CALL DISPLAYでは負の数を指定することはできません。ASCIIコード表を確認してください。負の数を出力したい場合は、超簡単命令セットの資料に従ってください。
「128以上はASCII出力できません」	CALL DISPLAYでは128以上を指定することはできません。ASCIIコード表を確認してください。
「使用できる数字は-128～127の整数で	入力している数字を確認してください。計

す」	算結果が-128～127の範囲を越えた場合にもこのエラー文が表示されます。
「使用できるのは256バイトまでです」	プログラムを256行以内にしてください。
「機械語は16文字もしくは24文字です」	機械語の入力を確認してください。また、機械語読み込みの際に0と1以外の文字や数字が混ざっている可能性があります。
「第二オペランドは00もしくは01で指定してください」	第二オペランドを確認してください。機械語読み込みの際に、CALLの後が00、01以外になっています。

5.4 ラベルに関してのエラー

「（ラベル名）は既に使用済みのラベルです」	既に同じラベル名があります。 ラベル名を変更してください。

5.5 オペランドに関してのエラー

「（オペコード）の第（数字）オペランドは〇〇です」という形式で表示されます。以下には〇〇ごとの対処方法を示します。

「レジスタです」	レジスタ以外のものがオペランドに指定されています。
「即値です」	即値以外のものがオペランドに指定されています。
「使用しません」	このオペコードはこのオペランドを使用しません。
「M(address)です」	M(address)以外のものがオペランドに指定されています。
「M(Rd)またはM(Rs+#n)です」	M(Rd)またはM(Rs+#n)以外のものがオペランドに指定されています。
「ラベルです」	label以外のものがオペランドに指定されています。
「DISPLAYかINPUTのいずれかです」	オペランドにDISPLAYかINPUTを指定してください。